Oracle ASMM (AMM) tips

Don Burleson

Note: ASMM and dynamic memory management has considerable overhead. See my important notes on Oracle dynamic memory management problems.
Automatic Memory Management has "issues" since its inception and by 11g release 2 it remains problematic, and in some cases ASMM should be disabled in 11g release 2. See MOSC note 793845.1 titled: "High direct path read waits in 11g" for complete details.

When using AMM you have to consider the interaction of these parameters: (according to Osama Mustafa)

- **sga_target: (pre 11g):** If the sga_target is set to some value then the automatic shared memory management (ASMM) is enabled, the sga_target value can be adjusted up to the sga_max_size parameter, not more than sga_max_size parameter value.
    - sga_max_size: The sga_max_size sets the overall amount of memory the SGA can consume but is not dynamic. The sga_max_size parameter is the max allowable size to resize the SGA memory area parameters.
- **memory_target (**starting in 11g): If memory_target is set, then AMM is enabled: If memory_target is set to non zero value and :
    - sga_target, sga_max_size and pga_aggregate_target are set to 0, then 60% of memory mentioned in memory_target is allocated to SGA and rest 40% is kept for PGA.
    - sga_target and pga_aggregate_target are set to non-zero values, then these values will be considered minimum values.
    - sga_target is set to non zero value and pga_aggregate_target is not set. still these values will be auto-tuned and pga_aggregate_target will be initialized with value of (memory_target-sga_target).
    - pga_aggregate_target is set and sga_target is not set. Both parameters will be auto-tuned. The sga_target will be initialized to a value of (memory_target-pga_aggregate_target).

memory_target	sga_target	sga_max_size	pga_aggregate_target	Behavior
non-zero	0	0	0	60% of memory_target to SGA, 40% to PGA
non-zero	non-zero		non-zero	Minimum values
non-zero	non-zero		un-set	pga_aggregate_target = memory_target - sga_target
non-zero	un-set		un-set	sga_target is set to memory_target - pga_aggregate_target

---

Oracle heuristic tuning is a well-known scientific approach that has been codified inside the Oracle 10g Automatic Shared Memory Manager (AMM) and Oracle Data Mining tools. Heuristic techniques are well-proven and accepted within the scientific community. The Heuristic approach is very straightforward. We observe our Oracle environment, searching for statistically-valid correlations, and apply these “rules of thumb” to new situations.

When Oracle9i first allowed "alter system" commands to morph the SGA, Oracle 10g

Search BC Oracle Sites
Search
Home
E-mail Us
Oracle Articles
SERVICES
Oracle Training
Oracle Tips
Oracle Forum
Class Catalog
SUPPORT
Remote DBA
Oracle Tuning
Emergency 911
RAC Support
Apps Support
Analysis
Design
Implementation
Oracle Support
CONSULTING
SQL Tuning
Security
Oracle UNIX
Oracle Linux
Monitoring
Remote support
Remote plans
Remote services
Application Server
Applications
Oracle Forms
Oracle Portal
App Upgrades
SQL Server
Oracle Concepts
Software Support
Remote Support
Development
Implementation
ABOUT BC
Consulting Staff
Consulting Prices
Help Wanted!
Oracle Books
Oracle Posters
Oracle Books
Oracle Scripts
Ion

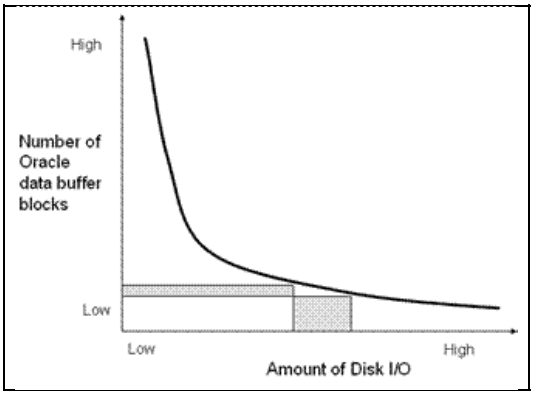
Excel-DB

Don Burleson Blog

introduced Automatic Memory Management, a reactive tool to re-size the RAM regions. While AMM is fine for smaller systems, there are reports of AMM causing performance problems:

- **They AMM to Please...Sometimes**
- **Oracle AMM resize operations can hurt performance**

The best example is the Oracle10g Automatic Shared Memory Manager (ASMM, or AMM) which observes historical buffer cache information and builds a statistically-reliable predictive model on re-sizing the data buffer. Using the rule-of-*thumb "Increase the data buffer cache size until the marginal benefit declines",* AMM can estimate the optimal cache size (the working-set of frequently-referenced data blocks).



When the marginal benefit of adding additional data blocks is plotted, the output looks like the following predictive model from *v$db_cache_advice (x$kcbcbh*).

When this data is plotted, the result is a typical 1/x curve as shown above. For an undersized buffer, a large reduction in disk I/O is achieved with a small increase in the size of a small RAM buffer.

In 11g and beyond, Oracle automatic memory management is configured using the *memory_target* and *memory_max_target* initialization parameters. The memory_target parameter specifies the amount of shared memory available for Oracle to use when dynamically controlling the SGA and PGA. The *memory_max_target* AMM parameter specifies the max size that *memory_target* may take. The docs note this on *memory_max_target*:

"For the MEMORY_MAX_TARGET initialization parameter, decide on a maximum amount of memory that you would want to allocate to the database for the foreseeable future.

That is, determine the maximum value for the sum of the SGA and instance PGA sizes."

Note: When using AMM, the values for the "traditional" pool parameters (*db_cache_size, shared_pool_size*, &c) are not ignored. Rather, they will specify the minimum size that Oracle will always maintain for each sub-area in the SGA.

Warnings about AMM

Quest Software's Guy Harrison has this warning about using the AMM:

"When you use MTS and AMM (or ASMM) together, PL/SQL programs that try to create large collections can effectively consume all available server memory with disastrous consequences . .

AMM allocates virtually all memory on the system to the large pool in order to accommodate the PL/SQL memory request. First it consumes the buffer cache, then it reduces the PGA_AGGREGATE_TARGET - all the way to zero!"

For more details on using automatic memory management (AMM), see:

http://www.dba-oracle.com/o10g_15.htm

http://www.dba-oracle.com/art_so_fav_10g.htm

How to disable AMM: See these important notes on disabling AMM (Automatic Space Memory Management)